

neither the plug-and-play manager nor the class driver become aware of masked devices and, hence, do not attempt to create disk objects for them.

To “add” back a LUN that was previously masked, the plug-and-play manager is initiated to 5 create and send a new data structure to the port driver 356 to be filled in. The plug-and-play manager is initiated by issuing from “user mode” a call to the filter driver 354, which itself issues a kernel mode `IO_INVALIDATE_DEVICE_RELATIONS` call. This causes the plug-and-play manager to issue calls (IRPs) to the port driver 356, which causes refill of the data structure. Then the filter driver 354 again intercepts the response from the port driver 356, and removes 10 any objects from the data structure that correspond to masked devices. Those skilled in art will appreciate that any other sequence of calls suitable for effecting refill of the data structure (e.g., `DEVICE_RELATIONS`) can be utilized.

To mask a LUN that is already available a command (i.e., `REMOVE`) is sent to the plug-and-play manager from “user mode” that identifies the device to be removed. The plug-and-play manager then removes all structures necessary for I/O (including disk objects). The filter driver 354 is active at all times to prevent any rescan from filling the data structure with a masked 15 device.

20 To unmask a LUN, a “remove” command (e.g., `CM_QUERY_AND_REMOVE_SUBTREE`) is issued to remove a device. Then a rescan is forced by opening the SCSI port drivers 356 and issuing to them a `CM_RENUMERATE_DEVNODE` command.

A further understanding of utilizing a device driver to mask LUNs in this alternate embodiment for a Windows™ 2000 host may be attached through the discussion that follows.

To mask LUNs at the SCSI port level an upper filter driver 354 to the SCSI port driver 356 is 5 used. The upper filter driver 356 catches Plug N Play request packets for devices on the SCSI port. The I/O request packet (IRP_MN_QUERY_DEVICE_RELATIONS) contains an array of all device objects attached to the SCSI port.

Using the first byte of the SCSI inquiry data, each device on the port is checked to make sure it is 10 a disk and then if the device is a disk queried for the LUN ID. If the device should be masked, the last device object in the array replaces the device object and the count of total devices is decremented. This effectively removes the masked device from the array. If the device is not masked the device remains in the list. After all masked disks have been removed the I/O request 15 packet is completed and the list is then sent back up to higher-level drivers. The masked disk devices are not visible to any driver higher than the filter driver 354. As a result, the SCSI class driver 352 does not make device objects for the masked devices, so the partitions on masked disks do not get mounted by the operating system.

The filter driver 354 does not change the SCSI port driver 356 data. Therefore, the SCSI port 20 driver 356 always has a list of all devices on its ports. The filter driver 354 simply prevents masked LUNs from being assigned.

Once Windows 2000 is booted care must be taken when masking out LUNs to avoid a surprise remove. When an unmasked LUN needs to be masked a user mode uninstall must be done to unmount the partitions and remove the disk safely from the plug-and-play manager. The SCSI bus is then rescanned and the device driver removes the device object from the array after a user mode uninstall of the disk has been completed successfully.

When a masked LUN needs to be unmasked the SCSI bus is rescanned. This unmasks the LUN since the device driver is not removing the device object from the array. Then the I/O request packet is completed which causes the SCSI class driver 352 to claim the disk and mount the partitions that reside on the disk.

Since the device driver is an upper filter driver 354 to the SCSI class driver 352, any host bus adapters that use the SCSI protocol work with this configuration. Fiber channel is an example of an adapter that uses SCSI protocol.

15

Association of LUN ID with Physical Device Object Name

As evident throughout the discussion above, the SAN manager 20 and agents 40 utilize the LUN IDs as identifiers for the storage devices (LUNs). Thus, by way of non-example, as discussed in the preceding sections, the disk manager 76 assigns LUNs to the hosts by loading their respective filter drivers 354 with the corresponding LUN IDs. The hosts are permitted to access LUNs whose LUN IDs are contained in the driver tables and are precluded from accessing the other LUNs.